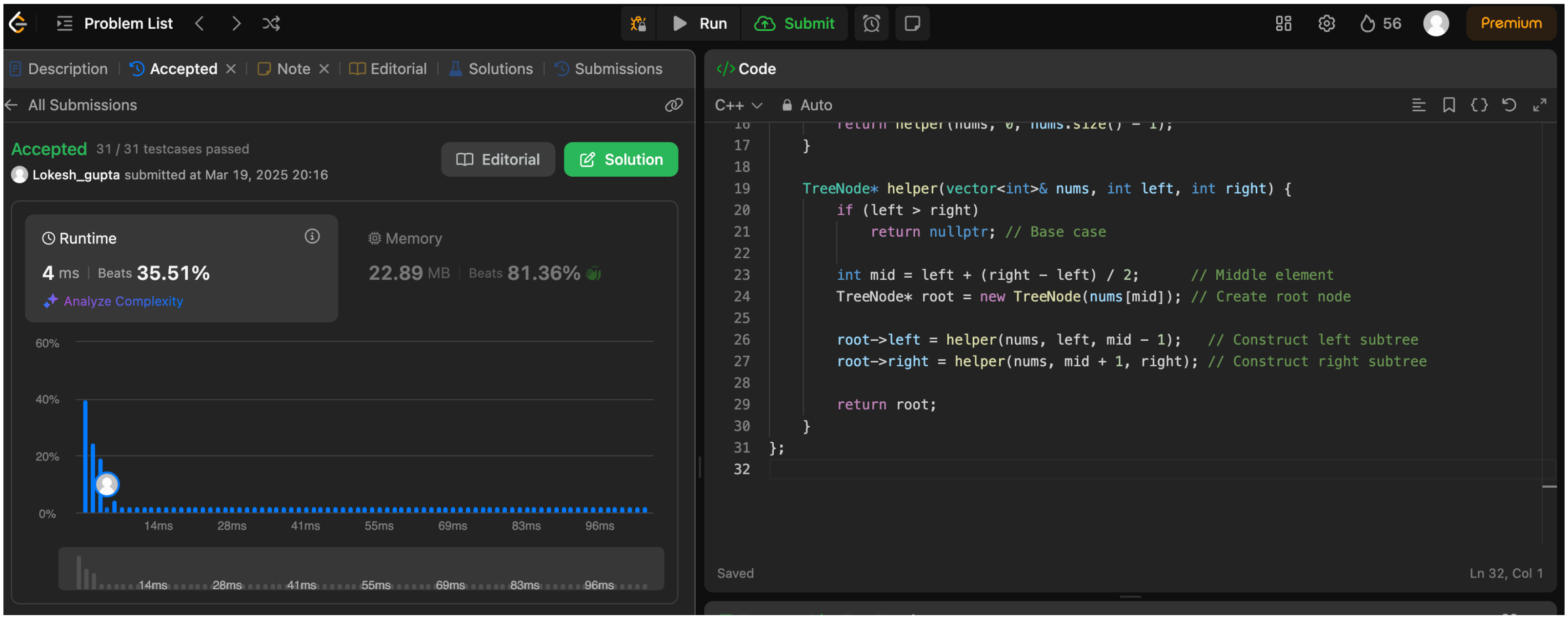
Experiment 6

Student Name: Lokesh Gupta **UID: 22BCS16079**
Branch: BE-CSE **Section/Group:FL_IOT-602/A**
Semester: 6th **Date of Performance:11/03/25**
Subject Name: Advanced Programming Lab-2 **Subject Code: 22CSP-351**

1. Implementation/Code:

Convert Sorted Array to Binary Search Tree

```
class Solution {
public:
   TreeNode* sortedArrayToBST(vector<int>&
nums) {
       return helper(nums, 0, nums.size() - 1);
   }

   TreeNode* helper(vector<int>& nums, int left,
int right) {
       if (left > right)
           return nullptr;
 int mid = left + (right - left) / 2;      // Middle
element
       TreeNode* root = new
TreeNode(nums[mid]); // Create root node

       root->left = helper(nums, left, mid - 1);   //
Construct left subtree
       root->right = helper(nums, mid + 1,
right); // Construct right subtree

       return root;
   }
};
```



Accepted 31 / 31 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 20:16

Runtime 4 ms | Beats 35.51%

Memory 22.89 MB | Beats 81.36%

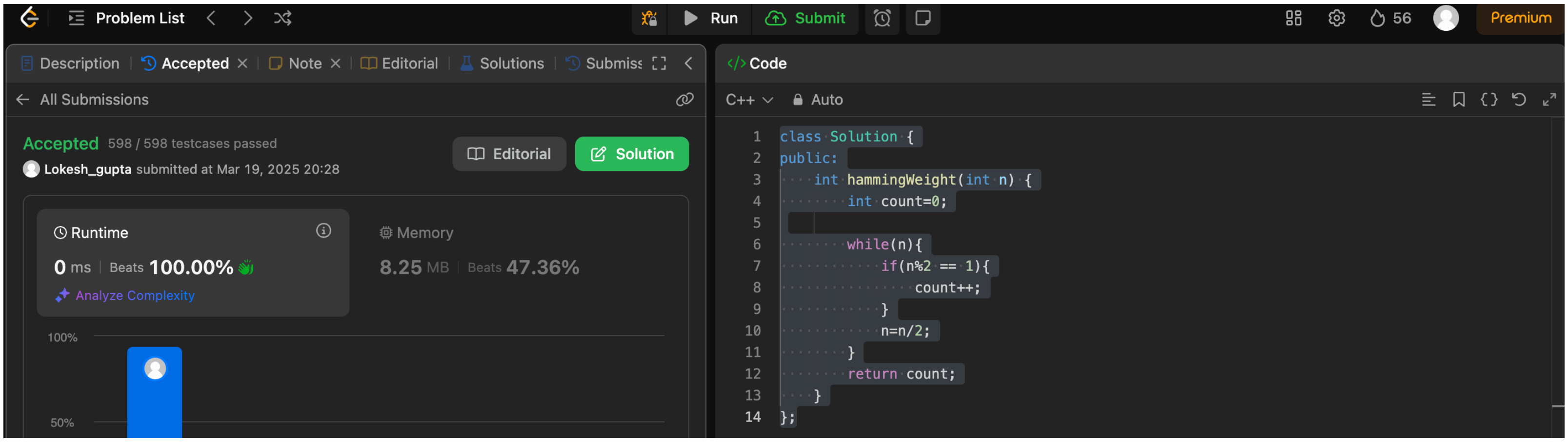
```
        return helper(nums, 0, nums.size() - 1);
    }

    TreeNode* helper(vector<int>& nums, int left, int right) {
        if (left > right)
            return nullptr; // Base case

        int mid = left + (right - left) / 2;      // Middle element
        TreeNode* root = new TreeNode(nums[mid]); // Create root node

        root->left = helper(nums, left, mid - 1);   // Construct left subtree
        root->right = helper(nums, mid + 1, right); // Construct right subtree

        return root;
    }
};
```

Saved

Ln 32, Col 1

Number of 1 Bits

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=0;

        while(n){
            if(n%2 == 1){
                count++;
            }
            n=n/2;
        }
        return count;
    }
};
```



Accepted 598 / 598 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 20:28

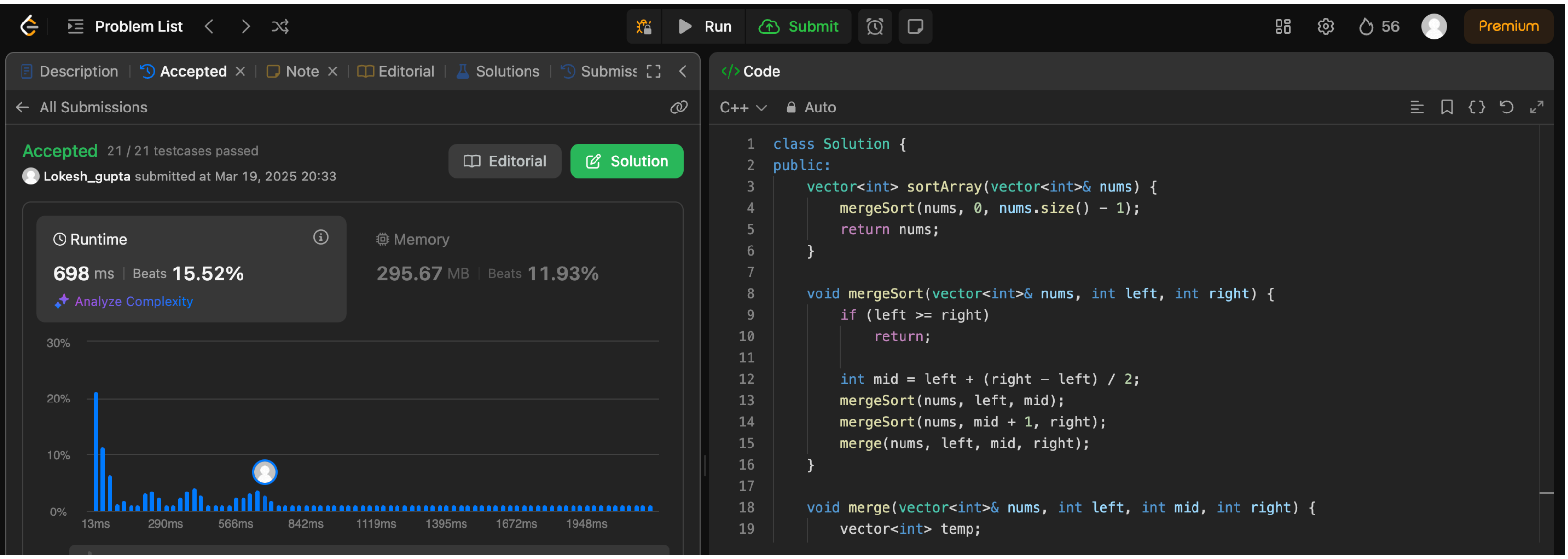
Runtime: 0 ms | Beats 100.00%

Memory: 8.25 MB | Beats 47.36%

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=0;

        while(n){
            if(n%2 == 1){
                count++;
            }
            n=n/2;
        }
        return count;
    }
};
```

Sort an Array

```
class Solution {
public:
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }

    void mergeSort(vector<int>& nums, int left, int right) {
        if (left >= right)
            return;
…          while (j <= right)
            temp.push_back(nums[j++]);

        for (int k = 0; k < temp.size(); k++)
            nums[left + k] = temp[k];
    }
};
```



Accepted 21 / 21 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 20:33

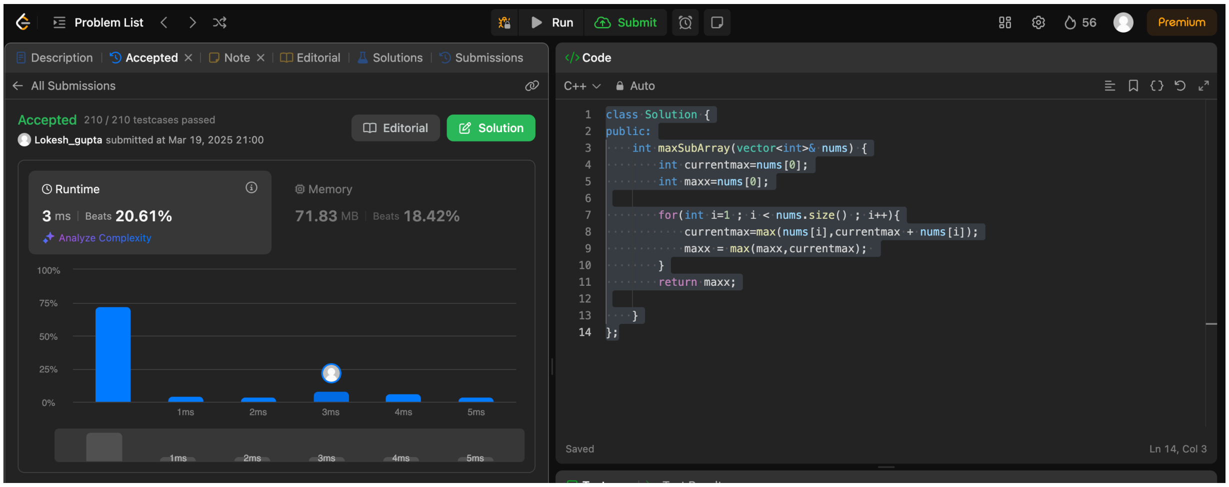
Runtime 698 ms | Beats 15.52%

Memory 295.67 MB | Beats 11.93%

```
class Solution {
public:
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }

    void mergeSort(vector<int>& nums, int left, int right) {
        if (left >= right)
            return;

        int mid = left + (right - left) / 2;
        mergeSort(nums, left, mid);
        mergeSort(nums, mid + 1, right);
        merge(nums, left, mid, right);
    }

    void merge(vector<int>& nums, int left, int mid, int right) {
        vector<int> temp;
```

Maximum Subarray

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int currentmax=nums[0];
        int maxx=nums[0];

        for(int i=1 ; i < nums.size() ; i++){
            currentmax=max(nums[i],currentmax + nums[i]);
            maxx = max(maxx,currentmax);
        }
        return maxx;

    }
```



Accepted 210 / 210 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 21:00

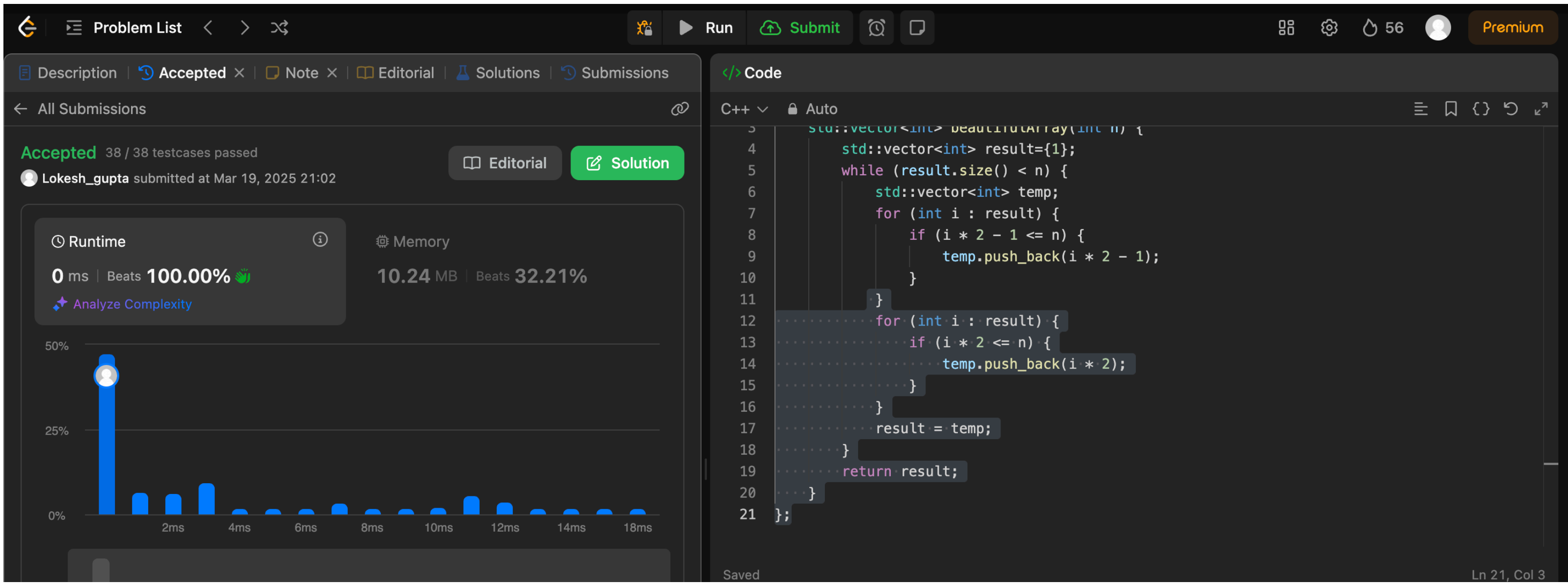
Runtime 3 ms | Beats 20.61%

Memory 71.83 MB | Beats 18.42%

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int currentmax=nums[0];
        int maxx=nums[0];

        for(int i=1 ; i < nums.size() ; i++){
            currentmax=max(nums[i],currentmax + nums[i]);
            maxx = max(maxx,currentmax);
        }
        return maxx;

    }
};
```

Beautiful Array

```
class Solution {
public:
    std::vector<int> beautifulArray(int n) {
        std::vector<int> result={1};
        while (result.size() < n) {
            std::vector<int> temp;
            for (int i : result) {
                if (i * 2 - 1 <= n) {
                    temp.push_back(i * 2 - 1);
                }
 }
            for (int i : result) {
                if (i * 2 <= n) {
                    temp.push_back(i * 2);
                }
            }
            result = temp;
        }
        return result;
    }
};
```



Accepted 38 / 38 testcases passed
Lokesh_gupta submitted at Mar 19, 2025 21:02

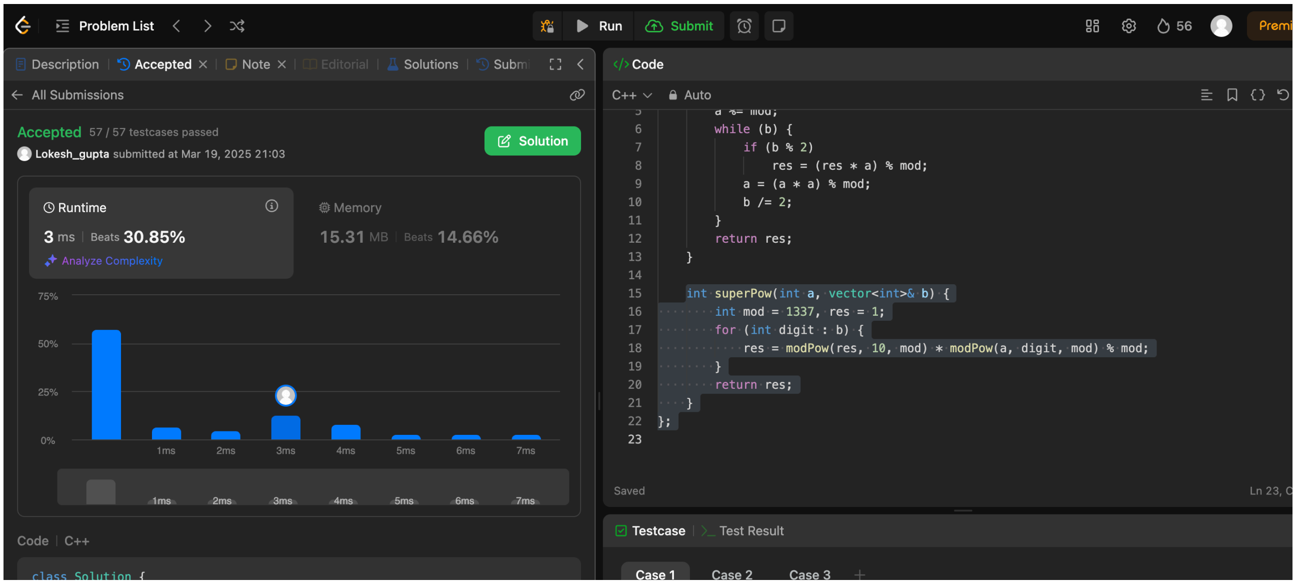
Runtime
0 ms | Beats 100.00%

Memory
10.24 MB | Beats 32.21%

```
    std::vector<int> beautifulArray(int n) {
        std::vector<int> result={1};
        while (result.size() < n) {
            std::vector<int> temp;
            for (int i : result) {
                if (i * 2 - 1 <= n) {
                    temp.push_back(i * 2 - 1);
                }
            }
            for (int i : result) {
                if (i * 2 <= n) {
                    temp.push_back(i * 2);
                }
            }
            result = temp;
        }
        return result;
    }
};
```

Super Pow

```
class Solution {
public:
    int modPow(int a, int b, int mod) {
        int res = 1;
        a %= mod;
        while (b) {
            if (b % 2)
                res = (res * a) % mod;
            a = (a * a) % mod;
            b /= 2;
        }
        return res;
    }
int superPow(int a, vector<int>& b) {
        int mod = 1337, res = 1;
        for (int digit : b) {
            res = modPow(res, 10, mod) * modPow(a, digit, mod) % mod;
        }
        return res;
    }
};
```



Accepted 57 / 57 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 21:03

Runtime: 3 ms | Beats 30.85%

Memory: 15.31 MB | Beats 14.66%

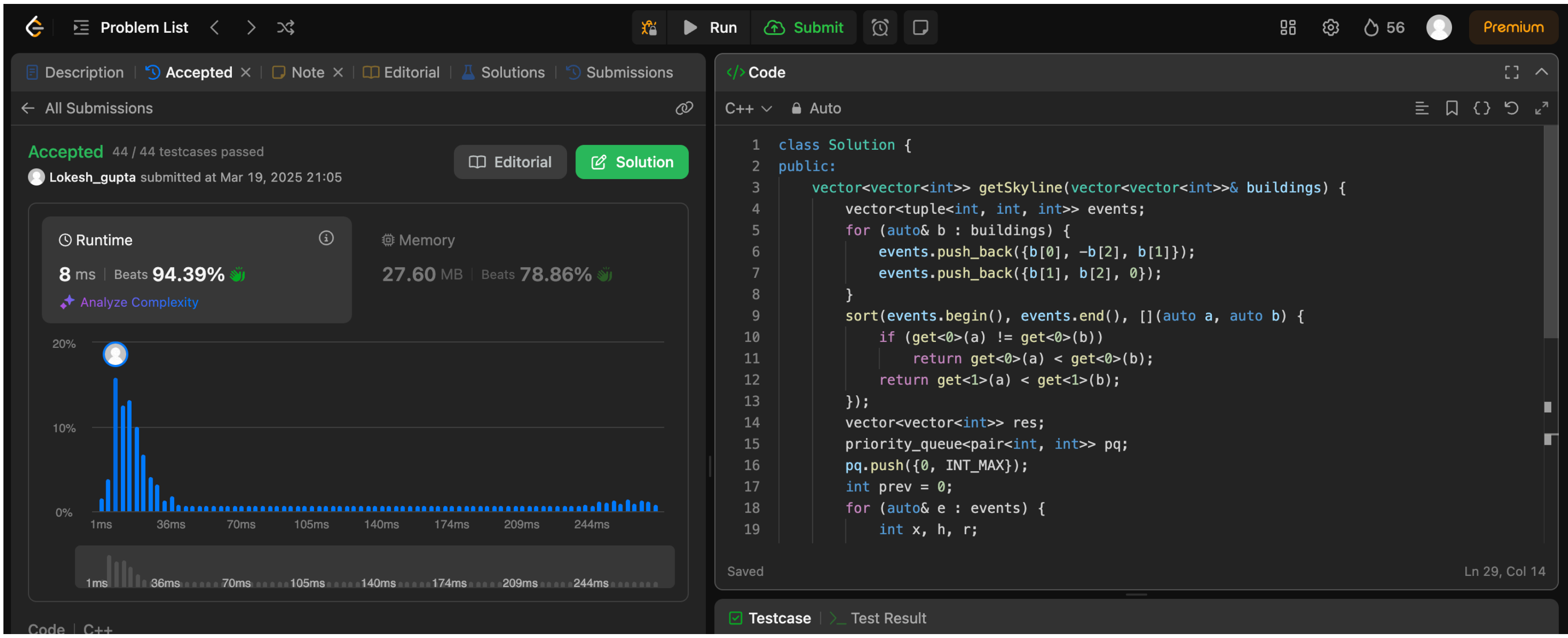
Code | C++

```
class Solution {
```

```
        a %= mod;
        while (b) {
            if (b % 2)
                res = (res * a) % mod;
            a = (a * a) % mod;
            b /= 2;
        }
        return res;
    }

    int superPow(int a, vector<int>& b) {
        int mod = 1337, res = 1;
        for (int digit : b) {
            res = modPow(res, 10, mod) * modPow(a, digit, mod) % mod;
        }
        return res;
    }
};
```

The Skyline Problem

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<tuple<int, int, int>> events;
        for (auto& b : buildings) {
            events.push_back({b[0], -b[2], b[1]});
            events.push_back({b[1], b[2], 0});
        }
        sort(events.begin(), events.end(), [](auto a, auto b) {
            if (get<0>(a) != get<0>(b))
                return get<0>(a) < get<0>(b);
            return get<1>(a) < get<1>(b);
        });
        vector<vector<int>> res;
priority_queue<pair<int, int>> pq;
        pq.push({0, INT_MAX});
        int prev = 0;
        for (auto& e : events) {
            int x, h, r;
            tie(x, h, r) = e;
…          return res;
    }
};
```



Accepted 44 / 44 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 21:05

Runtime

8 ms | Beats 94.39%

Memory

27.60 MB | Beats 78.86%

Code | C++

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<tuple<int, int, int>> events;
        for (auto& b : buildings) {
            events.push_back({b[0], -b[2], b[1]});
            events.push_back({b[1], b[2], 0});
        }
        sort(events.begin(), events.end(), [](auto a, auto b) {
            if (get<0>(a) != get<0>(b))
                return get<0>(a) < get<0>(b);
            return get<1>(a) < get<1>(b);
        });
        vector<vector<int>> res;
        priority_queue<pair<int, int>> pq;
        pq.push({0, INT_MAX});
        int prev = 0;
        for (auto& e : events) {
            int x, h, r;
```

Saved

Ln 29, Col 14